

# Basic Tutorial on the Use of Simulink

## OVERVIEW

Simulink is a powerful system modeling tool which is included with the MATLAB software package. It allows the user to graphically model equations and analyze the results, so that systems can be better understood and "what-if" scenarios can be more readily explored. This particular tutorial is designed to familiarize the student with the basic tools needed to develop and run a Simulink model. Once these basic concepts are understood, the student will be prepared for other tutorials in this series which specifically address the issues of modeling dynamic systems. Students who are already familiar with the basics of Simulink may wish to jump directly to the other tutorials.

## TUTORIAL CONVENTIONS

The following conventions will be used in this tutorial and the others in this series to facilitate an understanding of the command sequences:

- A ";" is used to separate sequential commands or entries.
- The down arrow symbol ▼ indicates that a pull-down menu must be accessed to find the appropriate menu item.
- For Radio Buttons the symbol ☉ will be used to indicate a selection.
- Menu bar items, and keyboard strokes will be *italicized and bold*.
- Custom entered commands will be bold and enclosed in parenthesis i.e. "**type here**"

Note: The input boxes within parameter windows always require numerical input. For this tutorial however, you may see variables in these input boxes to aid in understanding what values are being entered. You must input the numerical value associated with that variable for your model to run properly.

## GETTING STARTED

To begin, some of the basics of using Simulink are discussed.

### *Opening a Simulink Model*

Double click on the MATLAB icon on your desktop. The command prompt window will appear as in Fig. 1.

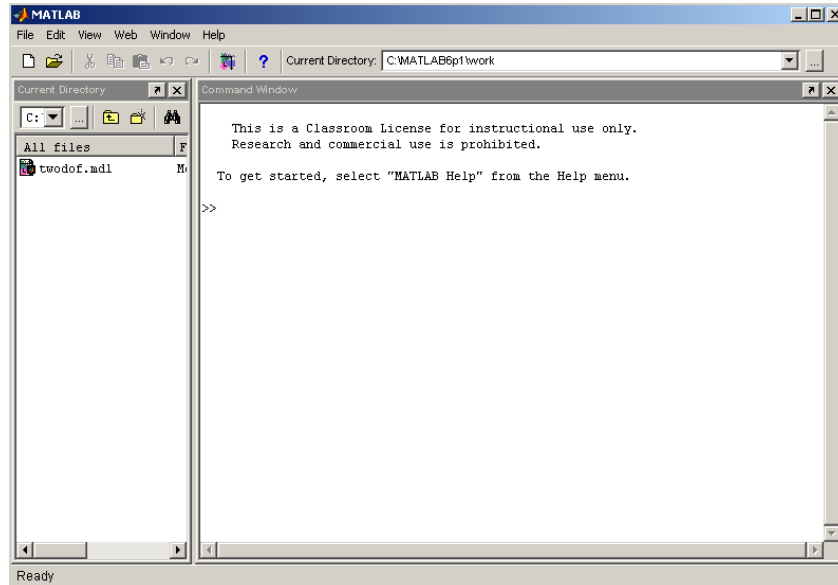


Fig. 1. The MATLAB command window.

At the prompt, type *simulink* → *enter*. A Simulink library browser appears as shown in Fig. 2.

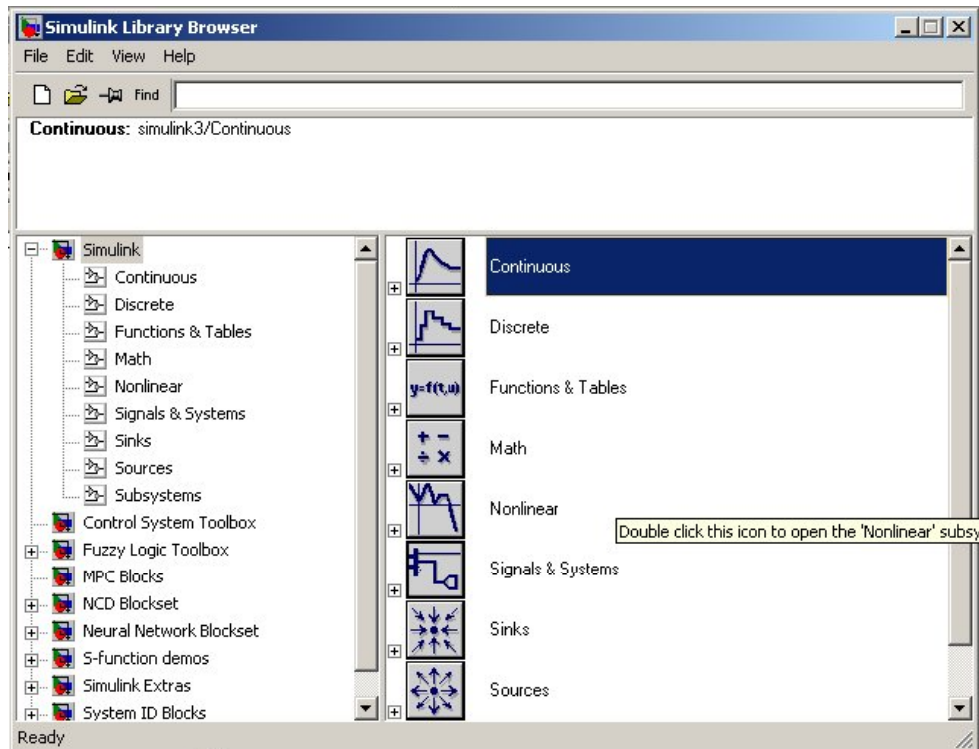


Fig. 2. The Simulink library browser.

From the *File* menu, select *new* → *model*. A new untitled window will appear. This will be the model space to construct the Simulink model.

## Bringing Blocks into the Model

To build a model, drag and drop blocks from the library into the model space. For example, to drag a Scope block into the model space, click on the Simulink library browser tree, then click on the **Sinks** submenu. A series of icons will appear on the right, each indicating a particular block type. Now click and drag the **Scope** icon onto the model space. Refer to Fig. 3 below.

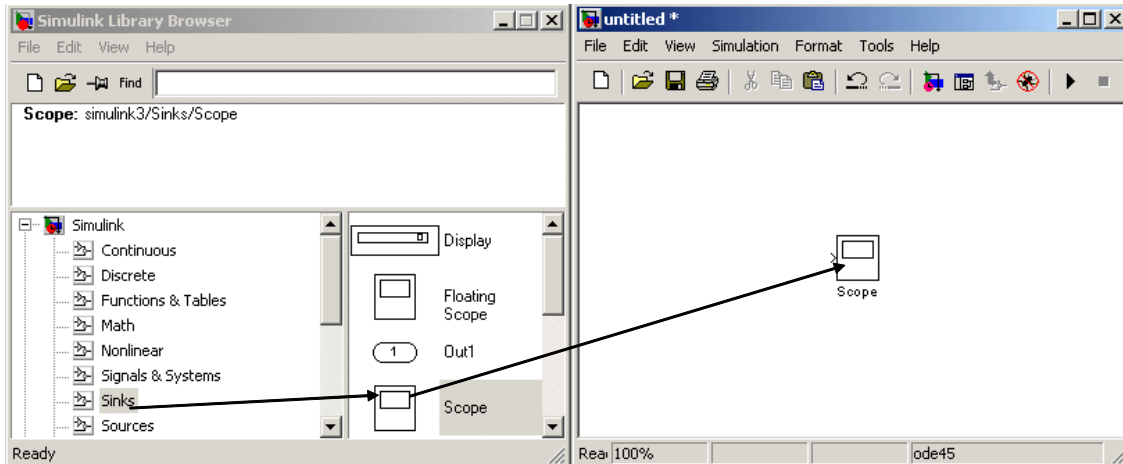


Fig. 3. Sequence for dragging and dropping Simulink blocks.

Other blocks can be accessed from the Simulink Library Browser. A summary of the blocks most commonly used and the submenu in which they can be found are in the table at the end of this tutorial.

## Connecting and Disconnecting Blocks

Lines are used to connect blocks, allowing "signal transmission" between blocks.

Drag and drop a **Sine Wave** block from the **Sources** submenu into the model space. Notice the output of the Sine block (the "greater-than" symbol on the right side) and the input of the Scope block (the "greater-than" symbol on the left side) in Fig. 4 below.



Fig. 4. Blocks prior to connection.

Left-click on the output port of the **Sine-Wave** block (the pointer will become a cross-hair), drag the line towards the input terminal of the next block, as shown in Fig. 5.

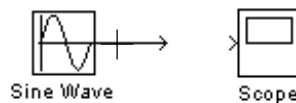


Fig. 5. Click and drag line.

When you reach the input port of the next block the pointer will become a double cross-hair. Release the left mouse button. The connected blocks will appear as in Fig. 6.

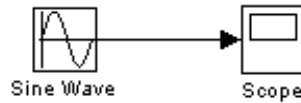


Fig. 6. Blocks attached.

Note: It is also possible to start with your cursor at the input port of a box and draw a line to the output port of another box.

### *Creating Branch Lines*

You can also create a branch line off an existing signal line. To do this, press and hold the **Ctrl** key, position the cursor on the line where the branch is to be created, then left-click and draw the branch line as in Fig. 7.

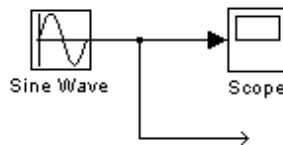


Fig. 7. Creating a branch line.

### *Changing Block Directions*

Sometimes it will become necessary to rotate a block. This can be done easily by right-clicking on the block, selecting **Format** → **Flip Block** or **Rotate Block**. The results of these choices can be seen in Fig. 8 below.

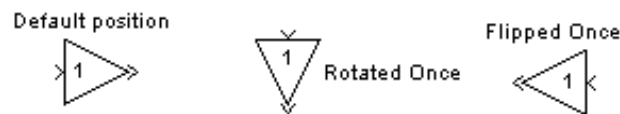


Fig. 8. Possible block positions.

### *Copying and Moving Blocks*

To create additional instances of a block, simply hold down the **Ctrl** key, click on the block, and drag off a copy. To move a block, simply left-click on the block and drag the block to a new location.

### Changing block labels and inserting text

To change block labels, simply click on the label and edit. To add a text box to the workspace, double-click anywhere on the workspace and a textbox will appear. Type into the textbox as usual; when done, click anywhere outside the textbox.

### Changing Block Parameters

Block parameters are used to control block input and output. The parameter window for any block can be viewed by double-clicking on the block. For example, double-clicking on a **Sum** block, examples of which are shown in Fig. 9, will open the “Block Parameters: Sum” window shown in Fig. 10 below.

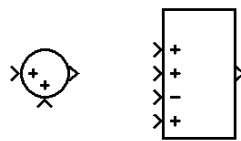


Fig. 9. Two examples of sum blocks.

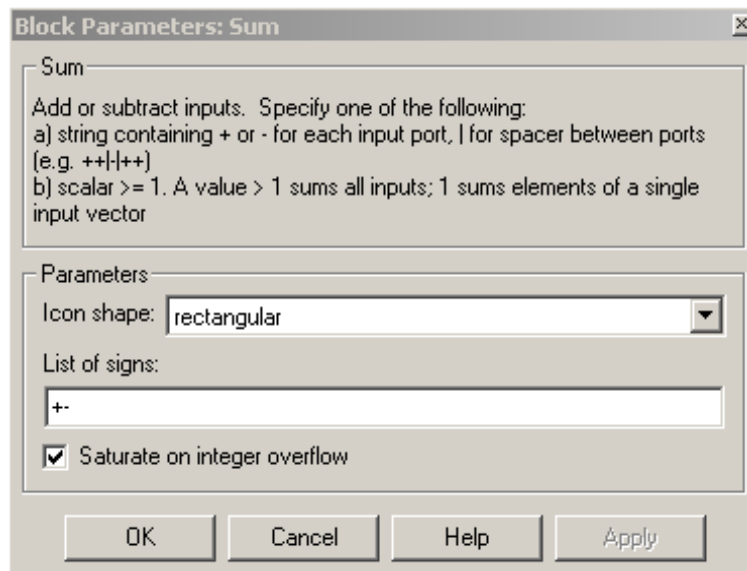


Fig. 10. Sum block parameter window.

Here, it is possible to change the shape of the icon and the number of input ports. Also, the input ports can be adjusted to act as a summer ( + ) or a comparator ( - ). After changing the block parameters as required select **Apply** → **OK** to activate the new parameters.

### Running the Simulation

Once the block parameters have been specified, it is time to run the model simulation. Select **Simulation** from the menu bar, then **Simulation Parameters**. The window shown in Fig. 11 will appear.

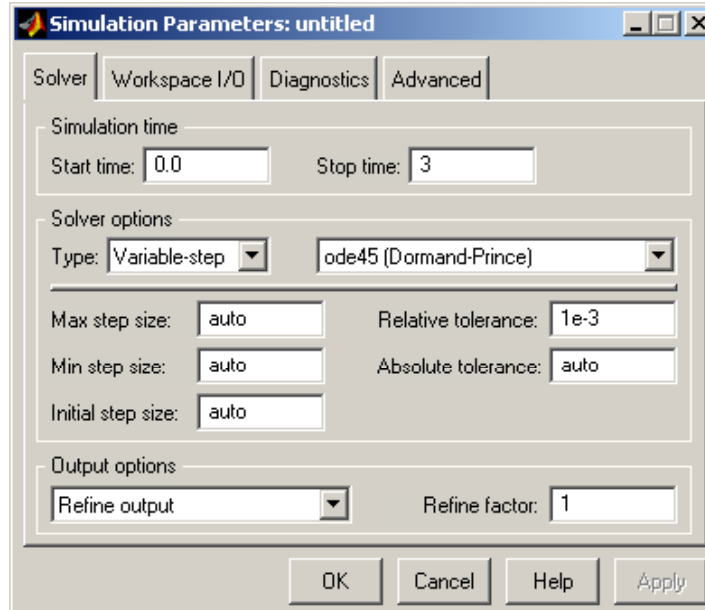


Fig. 11. Simulation parameters.

In general, ode45 is the standard solver and will be sufficient for most of the simulations performed. For most cases, the start time should be left at the default of 0; the stop time, however, should be adjusted as required to see the system response. If the stop time is set too short, the full response of the system may not be seen. If the time is set too long, then unnecessary data is being calculated. For large models this can add considerably to the simulation run time.

For the remaining fields, keep the default values. Once the fields have been updated select **Apply** → **OK**.

To run the simulation, go to the menu bar and select the ► button, as shown in Fig. 12 below.



Fig. 12. Simulation button viewing block output.

### ***Viewing the Simulation Results***

To view the output of any block, simply drag and drop a scope block into the model space (as shown in Fig. 3) and tap off of the line of interest. After running the simulation, the results can be viewed by double-clicking on the Scope block. A window like that in Fig. 13 will appear.

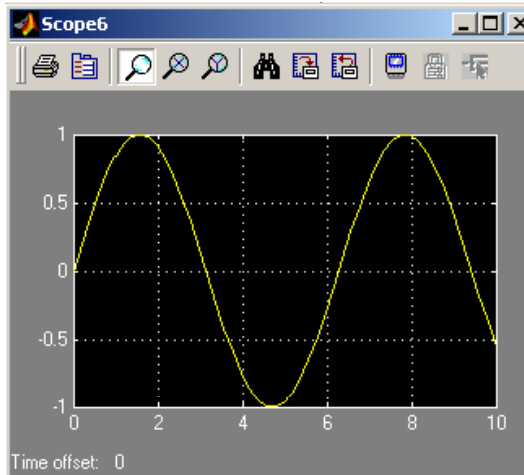


Fig. 13. Typical scope window.

Click on the *Binocular* icon at the top of the window to auto-scale the plot. You can also zoom in and out on the signal by selecting the appropriate buttons on the menu bar.

This concludes the introduction to Simulink. The basic skills necessary to construct some simple dynamic systems models, run these models, and analyze the results have been discussed. To continue developing skills, try some of the other related tutorials.